

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Infotehnoloogia õppekava

**Martin Merisalu**

# **Veebilehe loomise e-kursus gümnasistidele**

**Bakalaureusetöö (6 EAP)**

Juhendaja: MSc Ljubov Jaanuska

Tartu 2016

## Veebilehe loomise e-kursus gümnasistidele

### Lühikokkuvõte

Bakalaureusetöö eesmärk on õppematerjalide koostamine e-kursusele “Veebilehtede loomine”. Veebikeskkonda on lisatud kaks moodulit: HTML ja CSS. Materjali sisu ja ülesannete koostamine nõudis olemasolevate materjalide uurimist, sobivate meetodite valimist ning sünteesi, kuna kursuse sihtrühmaks on gümnasistid. Pärast e-kursuse läbimist oskavad õppijad koostada veebidokumente koos kujundamisega.

### Võtmesõnad

E-kursus, veebidisain, CSS, HTML, W3C.

### CERCS:

S281 Arvuti õpiprogrammide kasutamise metoodika ja pedagoogika

## Webpage e-course creation for high school students

### Abstract

The purpose of this thesis is to create studying materials for e-course “Web-design”. Two modules about HTML and CSS are available in the learning management system Moodle. The content of the materials and tasks required a wider research on available materials, selection of suitable methods and their synthesis since the target group of the e-course are high school pupils. On the completion of the models, the students are able to create a web page using HTML and CSS.

### Keywords

E-course, web design, CSS, HTML, W3C.

### CERCS:

S281 Computer-assisted education

# Sisukord

|       |  |    |
|-------|--|----|
| 1     | Sissejuhatus .....                                 | 5  |
| 2     | Kursuse ülevaade.....                              | 6  |
| 2.1   | Kursuse eesmärk .....                              | 6  |
| 2.1.1 | Olemasolevad kursused .....                        | 6  |
| 2.1.2 | Kursuse materjali allikad .....                    | 6  |
| 2.2   | Kursuse ülesehitus.....                            | 7  |
| 2.2.1 | Kursusel kasutatavad materjalid ning meetodid..... | 7  |
| 2.2.2 | Kursuse hindamine .....                            | 8  |
| 3     | Ülevaade HTML-i dokumendist .....                  | 9  |
| 3.1   | HTML-i dokumendi sünd .....                        | 9  |
| 3.2   | HTML-i dokumendi loomine.....                      | 9  |
| 3.2.1 | Dokumenditüüp.....                                 | 9  |
| 3.2.2 | HTML-i süntaks ning õigekiri .....                 | 10 |
| 3.2.3 | HTML-i dokumendi struktuur .....                   | 11 |
| 3.3   | HTML-i tekstitöötlus .....                         | 11 |
| 3.3.1 | HTML-i pealkirjad .....                            | 12 |
| 3.3.2 | HTML-i paragrahvid .....                           | 12 |
| 3.4   | HTML-i loendid .....                               | 12 |
| 3.4.1 | Järjestatud loend .....                            | 13 |
| 3.4.2 | Järjestamata loend .....                           | 13 |
| 3.4.3 | Definitsioonide loend .....                        | 14 |
| 3.5   | HTML-i tabelid .....                               | 14 |
| 3.6   | HTML-i hüperlingid .....                           | 15 |
| 3.7   | HTML-i vormid ja nupud.....                        | 15 |
| 3.7.1 | Vormi tekstiväli.....                              | 16 |
| 3.7.2 | Vormi valikukastid .....                           | 16 |
| 3.7.3 | Vormi nupud.....                                   | 17 |
| 3.8   | HTML-i valideerimine .....                         | 17 |
| 4     | CSS-i ülevaade .....                               | 18 |
| 4.1   | CSS-i ajalugu .....                                | 19 |
| 4.2   | CSS-i süntaks.....                                 | 19 |
| 4.2.1 | Elemendi ID ja Class atribuudid .....              | 20 |
| 4.2.2 | CSS-i koodi kommenteerimine .....                  | 20 |
| 4.2.3 | CSS-i parameetri väärtused .....                   | 20 |

|       |   |    |
|-------|---|----|
| 4.3   | CSS-i tekstitöötlus.....                  | 21 |
| 4.3.1 | CSS-i töö tekstiga .....                  | 21 |
| 4.3.2 | CSS-i tabelid .....                       | 22 |
| 4.4   | CSS-i kursorid.....                       | 23 |
| 4.5   | CSS-i pseudoklassid ning -elemendid ..... | 23 |
| 4.6   | CSS-iga positsioneerimine .....           | 24 |
| 4.7   | CSS-i taustad.....                        | 25 |
|       | Kokkuvõte.....                            | 28 |
|       | Viited .....                              | 29 |
|       | Litsents .....                            | 31 |

# 1 Sissejuhatus

Eesti on maailmas tuntud kui hea infoühiskonnaga riik. Kahjuks esineb ka puudusi. Üheks valupunktiks on praegune haridussüsteem ning ainekavad. Eesti haridus on täna seisus, kus mitte keegi pole rahul – ei õpetajad, õpilased ega ühiskond. Õpetajate peamiseks probleemiks on liigne töökoormus ja vähene palk. Õpilastele põhjustab pahameelt teoreetilisus ja vananenud teadmiste ammutamine, millest ei pruugi enam tulevikus kasu olla [1].

Uus haridusreform seab gümnaasiumis üheks tähtsamaks aineks matemaatika [2]. Selleks, et riigis oleks ka tulevikus jätkusuutlik infoühiskond, on meil vaja juurde just reaallaineid õppivaid noori. Informaatika ja inseneriteaduse alustalaks on matemaatika. Kahjuks on matemaatika ja teised reaallained noorte seas populaarsust kaotamas. Huvi kadumise põhjusteks on nende ainete teoreetilisus, igavus ning õpilastel puudub kindel arusaam, miks neil on vaja seda õppida.

Õppimine peaks koolides olema huvitav ning kaasahaarav. Ainekavad tuleks muuta mitmekesisemaks, et õpilased saaksid laialdasema ülevaate enda valitud valdkonna kohta. Õppijatele on vaja näidata, et matemaatika ei piirdu ainult teoreemidega või infotehnoloogia ei ole ainult programmeerimine. Huvi äratamine erinevate ainete vastu peaks toimuma juba madalamas õppeastmes: gümnaasiumis või isegi põhikoolis.

Veebilehtede loomise õpetamine on hea võimalus rikastada koolides arvutiõpetuse õppeprogrammi. See ei nõua väga palju aega ega ressursse lisaks. Veebilehtede loomise algkursus annab õpilastele ülevaadet infotehnoloogia valdkonnast ning veebis toimuvast. Tänapäeval kasutatakse veebilehtesid pea igas valdkonnas, seega on nende loomise oskus perspektiivikas ka tulevikus.

Käesoleva lõputöö eesmärgiks on luua eestikeelne õppematerjal veebilehe loomiseks. Töö sisuline osa koosneb kolmest peatükist. Esimene osa keskendub õppemetoodikale ehk kuidas kavatsetakse kursust läbi viia ning milliseid õppematerjale selleks kasutatakse. Teises peatükis antakse ülevaate HTML-i olemusest ning võimalustest. Kolmas peatükk uurib, kuidas CSS-i abil HTML-i lehti disainida ning milliseid huvitavaid võimalusi see pakub oma veebilehe loomisel.

## 2 Kursuse ülevaade

Käesolevas peatükis antakse loodavast kursusest ülevaade, kus kirjeldatakse selle ülesehitust ning eesmärgi. Samuti räägitakse olemasolevatest materjalidest ja e-õppe keskkonnast, mille abil viiakse läbi suurem osa kursusest.

### 2.1 Kursuse eesmärk

Kursuse eesmärgiks on luua veebikeskkonnas Moodle kursus, mis annaks praktilised juhendid ning oskused luua veebilehte. Tänapäeval on olemas mitmeid õppematerjale, et õppida veebilehti looma, aga kui need on võõrkeelsed (näiteks inglise keeles), siis võib tekkida keelebarjäär. Lisaks keelele ei osata neid ka leida. Väga paljud olemasolevad materjalid ei jõua õpilasteni mitmel põhjusel. Üheks valupunktiks iseseisva õppimise juures on materjalide liigne keerukus. Tihtipeale on olemas huvi õppida veebilehe loomist, aga materjaliga lähemalt tutvudes kaob see huvi ära. Algajana on väga raske aru saada võõrkeelsetest terminitest ning kui materjali on teema kohta väga palju, siis ei osata kuskilt õppimisega peale hakata. Samuti on väga paljud veebikursused tasulised. Seepärast on bakalaureusetöö peamiseks eesmärgiks võetud luua tasuta eestikeelsete materjalidega veebikursus, mis tekitaks õpilastes võimalikult palju huvi ning motiveeriks luua oma veebilehte. Seda peamiselt viisil, et kogu õpitut saab kohe interaktiivselt ka läbi proovida.

#### 2.1.1 Olemasolevad kursused

Antud teema kohta, mis käsitleb veebilehe loomist algtasemel, on olemas mitmeid veebikursuseid, nagu näiteks „Codecademy“. Samuti annab väga hea ülevaate erinevate võimaluste kohta „W3Schools“. Eestikeelseid tasuta materjale on vähe. Olemasolevad kursused on algajatele liialt iseseisvat tööd nõudvad ning pealiskaudsed [26].

„Codecademy“ lahendus on väga populaarne just oma interaktiivse osa poolest. Kursus on püstitatud nii, et õppija saab esmalt lugeda natuke teooriat ja näiteid ning seejärel loetu ise järele proovida. Instruksioone järgides võimaldab interaktiivne kursus kohe ennast kontrollida. Järgmise teema juurde saab minna alles siis, kui kõik juhised on täidetud. Kui koodis on midagi valesti, pakub programm võimalikke vigu, mis kasutaja võis teha ning seejärel saab neid kohe parandada.

„W3Schools“ sisaldab endas väga palju inglisekeelseid materjale, mis annavad piisava ülevaate nii teooriast kui ka praktilisest poolest. Jällegi on võimalus ise kohe läbi proovida. „W3Schools“ poolt pakutavad materjalid sobivad ehk rohkem õppuritele, kellel on varasemalt olnud kokkupuude veebilehtede loomisega või kui soovitakse leida kindlale probleemile lahendust ning näidist.

#### 2.1.2 Kursuse materjali allikad

Antud bakalaureusetöös on autor võtnud eestikeelse veebikursuse loomisel eeskujuks just „Codecademy“ ja „W3Schools“. Peamine põhjus seisneb selles, et need programmid on väga

interaktiivsed. Samuti on mõlemad veebilehed[2, 3] mujal maailmas saavutanud väga suure populaarsuse.

## 2.2 Kursuse ülesehitus

Kursus on loodud Moodle'i keskkonda. Moodle on õpikeskkond, mis on üle maailma väga suure populaarsuse saavutanud. Üle maailma on kokku ligi 85 miljonit kasutajat ning üle 9,5 miljoni kursuse [4]. Samuti on Moodle levinud ka Eesti koolides. Näiteks Tartu Ülikoolis kasutab Moodle'i õpikeskkonda üle 32 000 õpilase ning see sisaldab vähemalt 3500 e-kursust [5]. Moodle on avatud lähtekoodiga e-õppe keskkond, mistõttu on see vabavaraline rakendus ja kõik saavad seda tasuta oma seadmes kasutada [5].

Käesoleva töö jaoks on valitud Moodle'i keskkond sellepärast, et see võimaldab luua tasuta e-kursusi, mis kontsentreeruvad interaktiivsele õppele. See õpikeskkond sisaldab endas väga palju võimalusi ja eeliseid:

- vabavaraline rakendus;
- mugav kasutada nii õppijatel kui õpetajatel;
- võimalus siduda väliseid andmebaase (LDAP);
- rohkelt mooduleid;
- kasutajaskond on väga suur.

Igaüks, kes oskab programmeerida, võib luua uusi mooduleid ning lisarakendusi. [6]

Moodle võimaldab luua e-kursuse, kuhu on võimalik lisada erinevaid materjale. Käesoleva töö raames saab valitud õpikeskkonnas luua uue kursuse. See on omakorda jagatud mooduliteks ehk teemadeks, mis sisaldavad vastavaid õppematerjale. Samuti on igal kursusel privaatne foorum, kus õppijatel on võimalik omavahel arutleda või abi küsida.

### 2.2.1 Kursusel kasutatavad materjalid ning meetodid

Loodava kursuse valmimisel on töö autor mõelnud nii õpetamismeetodite kui ka õppimismeetodite peale. Õpetamise meetodi puhul on rõhk sellel, et hoida õpilasi pidevalt aktiivsena ning et säiliks huvi antud teema vastu. Õppematerjalidena kasutatakse võimalikult vähe teoreetilist osa ning rohkem praktikat.

Autor kasutab enda loodaval kursusel peamiselt tekstilisi ning pildilisi materjale. Illustreerivate näidete puhul on kasutatud ka videomaterjale. Samuti leidub täiendavaks iseseisvaks lugemiseks viiteid võõrkeelsetele materjalidele. Kõike loetut ja nähtut on võimalik ise kohe järele proovida. Selleks on õppijatele loodud võimalus kasutada olemasolevat tekstiredaktorit internetis, mis laseb ühel pool sisestada HTML-i koodi ning teisel pool simuleerib kirjutatut veebivaatena.

### 2.2.2 Kursuse hindamine

Moodle'i õpikeskkond pakub mitmeid võimalusi õpitulemuste hindamiseks. Kursuse läbiviija võib ka ise otsustada, mille alusel ja mil viisil hinnata. Moodle'i keskkonnas saab lisada testi, mille lõpetamisel saab õpilane kohe oma tulemuse ja vead teada. Iga teema kohta on test ning kursuse lõpus koondtest. Samuti tuleb teha individuaalset tööd, mida kaasõpilased saavad retsenseerida. Lisaks tuleb teha erinevaid ülesandeid, mida õpetaja saab hinnata näiteks hindemaatriksiga või juhendiga.

Hindemaatriksi puhul tuleb esmalt sisestada kriteeriumid ning võimalikud punktid. Hindamiseks saab klikkida maatriksiruuutudele ja lisada punktid, mille Moodle ise vastavalt kokku liidab. Juhendi puhul tuleb samuti sisestada kriteeriumid ning punktid ja hindamisel lisatakse kommentaar ning saadud punktid, mille Moodle sarnaselt eelmise meetodiga kokku liidab [7].



## 3 Ülevaade HTML-i dokumendist

### 3.1 HTML-i dokumendi sünd

HTML lühend tuleneb inglisekeelsetest sõnadest *HyperText Markup Language* ehk eesti keeles tähendab see „hüpertexti märgistuskeel“. Hüpertexti märkimise keelega märgitakse üles hüpertexti dokumendid. Sisuliselt on HTML-i dokument tekstifail, milles kirjeldatakse sisu ja kujundus kokkuleppeliste märgenditega (*tag*) ning interpreteeritakse brauseri (veebilehitseja) abil [8].

Esimene HTML-i dokument loodi 1990. aasta detsembris, kui loodi WWW (*World Wide Web*) ehk veeb. Veeb on ülemaailmne arvutite võrgustik, kus need saavad omavahel suhelda standardi HTTP (*Hypertext Transfer Protocol*) abil [9]. Esimene HTML versioon sisaldas endas lihtsamaid märgendeid, kuid aja jooksul (vaata tabel 1) seda uuendati ning täiendati uute lahendustega.

Tabel 1. HTML versioonid [10].

| Versioon  | Aasta |
|-----------|-------|
| HTML      | 1991  |
| HTML 2.0  | 1995  |
| HTML 3.2  | 1997  |
| HTML 4.01 | 1999  |
| XHTML     | 2000  |
| HTML5     | 2014  |

Veebistandardid koostab W3C (*World Wide Web Consortium*). See on 1994. a. Tim Berners-Lee poolt loodud rahvusvaheline kommuun, mis töötab välja avatud standardeid, et tagada veebi ühtlane ning pikaajaline areng. W3C on HTTP ja HTML standardite hoidja [11].

### 3.2 HTML-i dokumendi loomine

HTML dokumendi loomiseks on vaja tekstiredaktorit. Pärast faili loomist tuleks see salvestada laiendiga **html** või **htm**. Tavaliselt kannab veebilehestiku pealeht nimetust **index.html**. Samuti on veeebilehitsejad programmeeritud ära tundma seda faili kui esilehte.

#### 3.2.1 Dokumenditüüp

Veebilehe loomisel tuleks esimese reana ära deklareerida dokumendi tüüp ( **<!DOCTYPE>**). Erinevate HTML-i versioonide kohta on olemas deklaratsioonid (vaata joonis 1). Tänapäeval kasutatakse kõige sagedamini **<!DOCTYPE html>**, mis ühtlasi tähistab ka kõige uuemat versiooni HTML5.

```

HTML 5
<!DOCTYPE html>

HTML 4.01
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

XHTML
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

**Joonis 1.** HTML dokumendi tüübi kirjeldus.

Dokumendi tüübi deklaratsioon ise ei ole märgend. See aitab aru saada, millise HTML-i versiooniga on vaja veebileht kirjutada, et veebilehitseja teaks, kuidas dokumenti lugeda ning õigesti kuvada.

### 3.2.2 HTML-i süntaks ning õigekiri

HTML-i dokument koosneb elementidest. Märgendid paigutatakse märkide „<“ ja „>“ vahele. Üldjuhul on elementidel algusmärgend ning lõpumärgend, mille vahele jääb sisu. Lõpumärgendi elemendi ette kirjutatakse kaldkriips (vaata joonis 2). Elemente, millel ei ole lõpumärgendit, nimetatakse tühjadeks elementideks (vaata joonis 3). Kui dokumendi kirjeldamiseks kasutatakse XHTML-i versiooni, siis selliste elementide märgendi lõppu kirjutatakse kaldkriips (vaata joonis 4). Lisaks võib elementidel olla veel atribuut (vaata joonis 5).

```
<element>Sisu</element>
```

**Joonis 2.** HTML-i element.

```
<body>
  <br>
</body>
```

**Joonis 3.** HTML-i tühi element.

```
<body>
  <br />
</body>
```

**Joonis 4.** XHTML-i tühi element.

```

<element id="id"></element>
<element id="id">
<a href="http://www.google.com">See on Google</a>

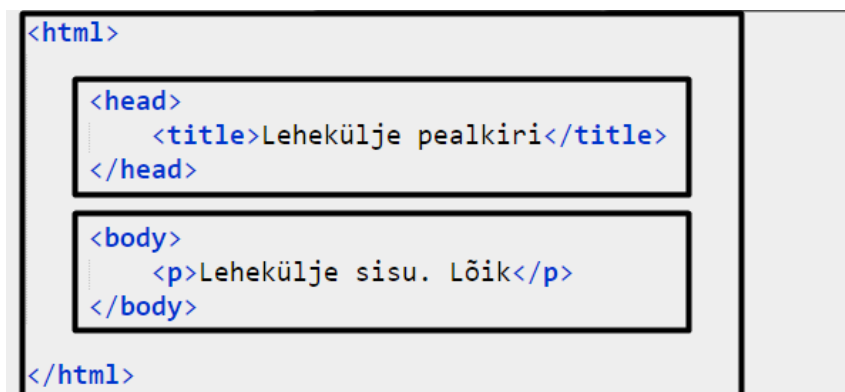
```

**Joonis 5.** HTML-i atribuut.

Mahukate veebilehtede loomisel on mõistlik koodi lugemine teha lugejale lihtsamaks. Soovitavalt võiks elemendid kirjutada dokumendis erinevatele ridadele. Otseselt ei ole see nõutud, sest veebilehitseja tõlgendab veebilehe sisu elementide kaupa. Teiseks on hea tava koodi kommenteerida. Selleks lisatakse kommentaari algusesse „<!--“ ning lõppu „-->“.

### 3.2.3 HTML-i dokumendi struktuur

Igal veebilehel on oma struktuur ja selle loomiseks kasutame märgendeid **<html>**, **<head>** ning **<body>** (vaata joonis 6).



Joonis 6. HTML-i dokumendi struktuur.

Pärast dokumendi tüübi deklaratsiooni (DTD) algab **<html>** element. See on HTML-i lehe juurelement ning oluline on ka elemendi lõppu tähistav märgend **</html>**. Veebilehitseja saab aru, kust alates dokument algab ning lõpeb. **<html>** elemendi sees kirjeldatakse ära kaks peamist osa: **<head>** ehk päis ning **<body>** ehk sisu või keha.

Päis sisaldab metaandmeid ehk andmeid andmete kohta. Sisuliselt kirjeldatakse ära dokumendis kasutatavad skriptid, stiililehed, võtmesõnad, dokumendi nimi (vaata joonis 6), jms. Elemendid, mida saab kasutada **<head>** sees on **<title>**, **<style>**, **<base>**, **<link>**, **<meta>**, **<script>** ja **<noscript>** [12]. Päise andmed veebilehitseja aknas ei ole nähtavad, välja arvatud sisu märgendite vahel, mis määrab dokumendi nime (vaata joonis 7).



Joonis 7. Veebilehe tiitel.

Keha osas kirjutatakse näha brauseri aknas. Teisisõnu kirjutatakse **<body>** elemendi sees kõike, mida soovitakse kuvada veebilehel: tekst, pildid, tabelid jms.

### 3.3 HTML-i tekstitöötlus

HTML-i dokumendi keha osas pannakse kirja kõik, mida tahetakse veebilehitsejas kuvada. Teksti kuvamiseks erineval moel on palju elemente (vaata tabel 2). Näiteks paksu kirja saamiseks kasutatakse märgendit **<b>** (**bold**), kursiivi jaoks **<i>** (*italic*) ning tavalise lõigu loomiseks **<p>** (paragrahv).

**Tabel 2.** Teksti vormingu märgised [13].

| Element               | Kirjeldus   |
|-----------------------|---|
| <b>&lt;b&gt;</b>      | Teksti tegemine paksuks                           |
| <b>&lt;em&gt;</b>     | Teksti rõhutamine                                 |
| <b>&lt;small&gt;</b>  | Teksti tegemine väiksemaks                        |
| <b>&lt;i&gt;</b>      | Teksti kursiiviks tegemine                        |
| <b>&lt;strong&gt;</b> | Teksti tegemine tugevaks                          |
| <b>&lt;sub&gt;</b>    | Tekstis allindeksi määramine                      |
| <b>&lt;sup&gt;</b>    | Tekstis ülaindeksi määramine                      |
| <b>&lt;del&gt;</b>    | Teksti maha tõmbamine/läbikriipsutamine           |
| <b>&lt;ins&gt;</b>    | Asendustekst pärast <b>&lt;del&gt;</b> kasutamist |
| <b>&lt;mark&gt;</b>   | Teksti markeerimine                               |

### 3.3.1 HTML-i pealkirjad

HTML-i dokumentides on oluline pöörata tähelepanu pealkirjadele. Kasutusel on kuue erineva tasemega elemendid **<h1>** kuni **<h6>**. Kuna otsingumootorid kasutavad pealkirjasid, et leida vasteid veebilehelt, siis ei tohiks kasutada kirja suurendamiseks või teksti paksuks tegemiseks pealkirja laade [14].

### 3.3.2 HTML-i paragrahvid

Element **<p>** tähistab HTML-i dokumendis paragrahve. Sisuliselt tähendab see lõiku, kuhu sisse saab lisada teksti, pilte, videoid jne (vaata joonis 8). Tavalist teksti saab HTML-i dokumendis ka niisama kirjutada, aga selle kujundamiseks (CSS abil) peaks kasutama paragrahvi elementi [8].



**Joonis 8.** Paragrahvi kasutamise näide.

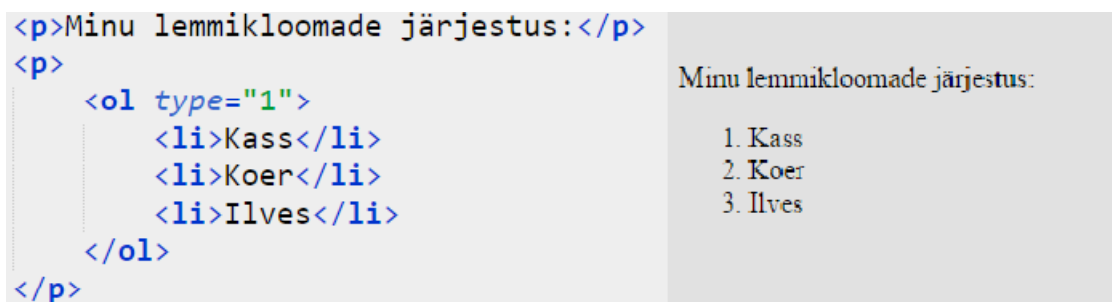
## 3.4 HTML-i loendid

Veebilehel kasutatakse järjestatud loendit (*ordered list*), järjestamata loendit (*unordered list*) ning definitsioonide loendit (*description list*). Neid saab kasutada mitme erineva otstarbe jaoks:

- loendi loomiseks, et teksti oleks parem mõista ning lihtsam jälgida;
- menüü või sisukorra süsteemi loomiseks;
- sõnaraamatulaadse loendi loomiseks.

### 3.4.1 Järjestatud loend

Järjestatud loendi loomisel kasutatakse märgendit `<ol>` (vaata joonis 9). Iga nimistus olev elemendi ette käib `<li>`. Samuti on võimalus märgendile `<ol>` lisada atribuut (vaata tabel 3), mis määrab loendi tüübi.



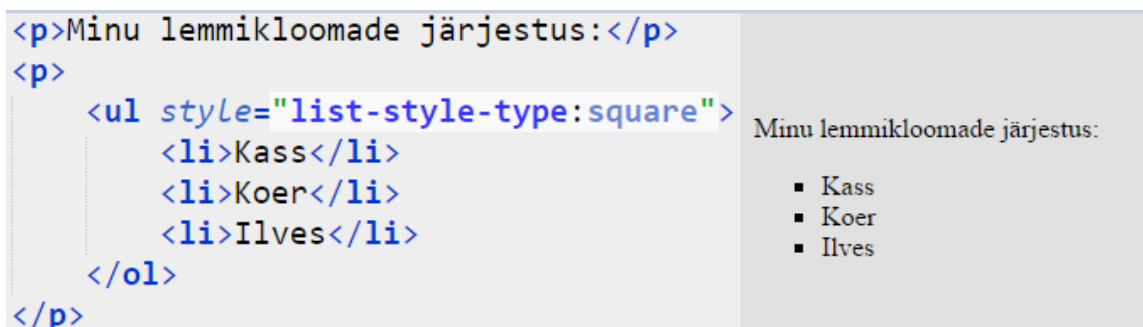
Joonis 9. Järjestatud loend.

Tabel 3. Järjestatud loendi tüübi atribuudid [15].

| Tüüp                  | Kirjeldus   |
|-----------------------|---|
| <code>type="1"</code> | Loendi elemendid on nummerdatud                             |
| <code>type="A"</code> | Loendi elemendid on järjestatud suurtähtkirja abil          |
| <code>type="a"</code> | Loendi elemendid on järjestatud väiketähtkirja abil         |
| <code>type="I"</code> | Loendi elemendid on järjestatud suurte roomanumbrite abil   |
| <code>type="i"</code> | Loendi elemendid on järjestatud väikeste roomanumbrite abil |

### 3.4.2 Järjestamata loend

Järjestamata loendeid luuakse elemendi `<ul>` (vaata joonis 10) abil ning nimistu ees kasutatakse märgendit `<li>`. Samuti on võimalus lisada ka atribuute (vaata tabel 4), et määrata ära nimekirja stiil.



Joonis 10. Järjestamata loend.

**Tabel 4.** Järjestamata loendi atribuudid [15].

| Stiil                         | Kirjeldus                              |
|-------------------------------|--|
| <b>list-style-type:disc</b>   | Loendi ees kasutatakse täppi           |
| <b>list-style-type:circle</b> | Loendi ees kasutatakse ringi           |
| <b>list-style-type:square</b> | Loendi ees kasutatakse ruutu           |
| <b>list-style-type:none</b>   | Loendi ees ei kasutata ühtegi elementi |

### 3.4.3 Definiitsioonide loend

Definiitsioonid loendi loomiseks (vaata joonis 11) kasutame järgmisi elemente:

- `<dl>` - definiitsioonide loendi loomine;
- `<dt>` - defineeritav termin;
- `<dd>` - definiitsioon ise.

|   |  |
|---|--|
| <pre> &lt;p&gt;Sõnaraamat:&lt;/p&gt; &lt;dl&gt;   &lt;dt&gt;HTML dokument&lt;/dt&gt;   &lt;dd&gt;hüpertext-dokument&lt;/dd&gt;   &lt;dt&gt;CSS&lt;/dt&gt;   &lt;dd&gt;kaskaadlaadistik&lt;/dd&gt;   &lt;dd&gt;stiililehed&lt;/dd&gt; &lt;/dl&gt; </pre> | <p>Sõnaraamat:</p> <p>HTML dokument<br/>hüpertext-dokument</p> <p>CSS<br/>kaskaadlaadistik<br/>stiililehed</p> |
|---|--|

**Joonis 11.** Definiitsioonide loend.

## 3.5 HTML-i tabelid

HTML võimaldab luua erinevaid tabeleid. Tabel on määratud märgisega **<table>**, mis omakorda jaguneb ridadeks (**<tr>** ehk *table row*) ja read omakorda lahtriteks (**<td>** ehk *table data*). Samuti võib tabeli read jagada tabeli päisteks (**<th>** ehk *table header*). Elementid **<td>** määravad ära lahtri sisu, nagu näiteks tekst, pilt, loendid, jne. Tabeli pealkirja määramiseks kasutatakse märgendit **<caption>** (vaata joonis 12).[18]

```

<table border="1">
  <caption>Monitorid</caption>
  <tr>
    <th>Tootja</th>
    <th>Ekraani suurus</th>
    <th>Hind</th>
  </tr>
  <tr>
    <td>LG</td>
    ...
  </table>

```

| Monitorid |                |         |
|-----------|----------------|---------|
| Tootja    | Ekraani suurus | Hind    |
| LG        | 22"            | 123,35€ |
| Samsung   | 19"            | 101,62€ |
| NEC       | 23"            | 349,60€ |

**Joonis 19.** Tabeli loomine [19].

Tabeli kujundamine ning vormindamine pannakse kirja kaskaadlaadistiku failis.

### 3.6 HTML-i hüperlingid

Hüperlink ehk link on kasutusel selleks, et saaks liikuda ühelt veebilehelt teisele. See oli ühtlasi ka üks põhjus veebi idee sünniks. HTML-i dokumendis kasutatakse märgendit **<a>** (*anchor* ehk ankur), mille vahele lisatakse sisu ning millele peale vajutades link töötab. Selleks, et veebilehitseja oskaks suunata õigesse kohta, kasutame märgendi sees atribuuti **href**. See võib viidata erinevatele veebiressurssidele, nagu näiteks teine HTML-i dokument, fail, pilt jms (vaata joonis 12) [8].

```
<a href="http://www.w3schools.com/html/html_links.asp">W3Schools</a>
```

Joonis 12. Hüperlingi loomine.

HTML-i hüperlingile on võimalik lisada atribuuti *target* (vaata joonis 13), mille väärtus (vaata tabel 5) määrab ära, kuidas link avatakse[8].

```
<a href="http://www.w3schools.com/html/html_links.asp" target="_blank">W3Schools</a>
```

Joonis 13. Atribuudi *target* kasutamine.

Tabel 5. Atribuut *target* väärtused [15].

| Väärtus          | Kirjeldus                           |
|------------------|-------------------------------------|
| <b>_blank</b>    | Link avatakse uues aknas            |
| <b>_self</b>     | Link avatakse samas aknas           |
| <b>_parent</b>   | Link avatakse „ema“ raamistikus     |
| <b>_top</b>      | Link avatakse terves aknas          |
| <b>framename</b> | Link avatakse nimetatud raamistikus |

### 3.7 HTML-i vormid ja nupud

HTML-i vorminguid kasutatakse selleks, et koguda kasutaja sisendit ja saata need veebiserverile. Selle loomiseks on loodud märgend **<form>**. Vormi elemendid võivad olla erinevad sisendid, nagu näiteks sisestusväljad, raadio-nupud, valikvastustega kastid jms.[16]

Vormid koosnevad kahest osast. Esimene osa on see, mis on nähtav kasutajale ning teine osa andmeid töötlev veebiserver. HTML võimaldab meil luua ainult visuaalset osa.[17]

Vormi kõige tähtsam element on **<input>**. Sellele märgisele on võimalik lisada erinevaid atribuute (vaata tabel 6) [16].

Tabel 6. Elemendi **<input>** atribuudid [16].

| Väärtus         | Kirjeldus   |
|-----------------|---|
| <b>text</b>     | Tekitab tekstivälja   |
| <b>radio</b>    | Tekitab valikvariandid (kasutaja saab teha ühe valiku)                      |
| <b>checkbox</b> | Tekitab valikvariandid (kasutaja saab valida mitme variandi vahel)          |
| <b>submit</b>   | Defineerib nupu, et edastada info   |
| <b>action</b>   | Defineerib tegevuse, mis teostatakse pärast „ <i>submit</i> “ käsu täitmist |

Veebilehel võib olla korraga mitu vormi ning küsitavaid väärtusi või valikuid, mida on küsitud. Selleks, et serverisse saadetavatest andmetest kasu oleks, tuleks erinevatele elementidele

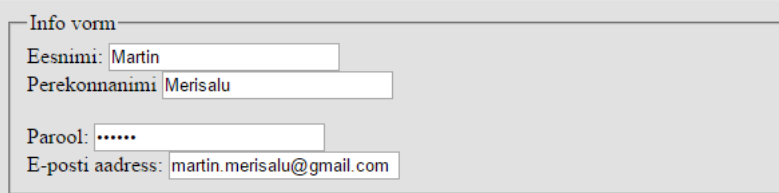
anda nimi. Seda saab teha atribuudiga **name**. Selleks, et grupeerida omavahel seotud elemendid vormis, kasutatakse märgendit **<fieldset>**. Sellele pealkirja andmiseks kasutatakse märgendit **<legend>**. Vormi koostamisel on oluline, et kasutaja teaks, milliseid andmeid temalt oodatakse. Selleks võimaldab HTML kasutada silte **<label>** (vaata joonis 14).

### 3.7.1 Vormi tekstiväli

HTML-i tekstiväli on määratud elemendi **<input>** atribuudiga **text**. See võimaldab sisestada kasutajal tähti, numbreid jms. Muid parameetreid määramata on kastil vaikimisi pikkus 20 sümbolit.

Tekstiväljad on ka väljad, mis küsivad näiteks parooli või e-posti aadressi (vaata joonis 14).

```
<form name="minuVorm">
  <fieldset>
    <legend>Info vorm</legend>
    <label>Eesnimi:</label>
    <input type="text" name="e_nimi"><br>
    <label>Perekonnanimi</label>
    <input type="text" name="p_nimi"><br>
    <br>
    <label>Parool:</label>
    <input type="password" name="parool"><br>
    <label>E-posti aadress:</label>
    <input type="email" name="epost">
  </fieldset>
</form>
```



The screenshot shows a web form with the title "Info vorm". It contains four input fields, each with a label to its left: "Eesnimi:" with the value "Martin", "Perekonnanimi:" with the value "Merisalu", "Parool:" with masked characters ".....", and "E-posti aadress:" with the value "martin.merisalu@gmail.com".

Joonis 14. Vormi tekstivälja näidis.

### 3.7.2 Vormi valikukastid

HTML pakub kahte erinevat meetodit valikukastide loomiseks. Ühel juhul kasutaja saab teha ainult ühe valiku. Selliseid valikukaste nimetatakse raadionuppudeks. Selleks on kasutusel **<input>** elemendil tüübikirjeldus **radio** (vaata joonis 15). Teisel juhul saab kasutaja teha valiku ühe või enama variandi osas. Selleks tuleb kasutada tüübikirjeldust **checkbox** (vaata joonis 15).



|  |   |
|--|---|
| <pre> &lt;form name="radio"&gt;   &lt;input type="radio" name="sugu" value="mees" checked&gt; Mees&lt;br&gt;   &lt;input type="radio" name="sugu" value="naine"&gt; Naine&lt;br&gt;   &lt;input type="radio" name="sugu" value="muu"&gt; Muu &lt;/form&gt; &lt;br&gt; &lt;form name="checkbox"&gt;   &lt;input type="checkbox" name="sugu2" value="kass" checked&gt; Kass&lt;br&gt;   &lt;input type="checkbox" name="sugu2" value="koer"&gt; Koer&lt;br&gt;   &lt;input type="checkbox" name="sugu2" value="muu"&gt; Muu &lt;/form&gt; </pre> | <input checked="" type="radio"/> Mees<br><input type="radio"/> Naine<br><input type="radio"/> Muu<br><br><input checked="" type="checkbox"/> Kass<br><input checked="" type="checkbox"/> Koer<br><input type="checkbox"/> Muu |
|--|---|

Joonis 15. Vormi valikukastid.

### 3.7.3 Vormi nupud

Selleks, et saata sisestatud andmeid serverisse, on vaja nuppu. HTML-is saab seda teha **submit** abil (vaata joonis 16).

```

<input value="registreeru" name="reg" type="submit">
<input value="puhasta väljad" name="clr" type="reset">

```

Info vorm

Eesnimi:

Perekonnanimi:

Parool:

E-posti aadress:

registreeru    puhasta väljad

Joonis 16. Vormi nupud.

## 3.8 HTML-i valideerimine

HTML-i dokumendi loomisel on paika pandud kindlad standardid. Need tagavad veebilehtede parema toimimise tõrgeteta erinevates brauserites. Selleks on W3C loonud HTML5 validaatori, mis võimaldab kogu koodi kontrollida, kas see vastab standarditele. Kui koodiga on kõik korras, kuvatakse vastav teade (vaata joonis 17). Vastasel juhul teavitab validaatori veast ning annab ka kätte juhised, kus viga esineb (vaata joonis 18).

Document checking completed. No errors or warnings to show.

Joonis 17. Validaatori läbinud kood.

**Error** Unclosed element `form`.

From line 9, column 1; to line 9, column 22

```

<body><form name="minuVorm"> </file>

```

Joonis 18. Validaator tuvastas vea.

Koodi valideerimine aitab hiljem veebilehte paremini hooldada. Lisaks aitab see ära hoida ebameeldivaid üllatusi veebilehe kuvamisel, kui midagi on paigast ära või ei kuva korrektselt. Samuti võib validaatorit kasutada koodi silumiseks.

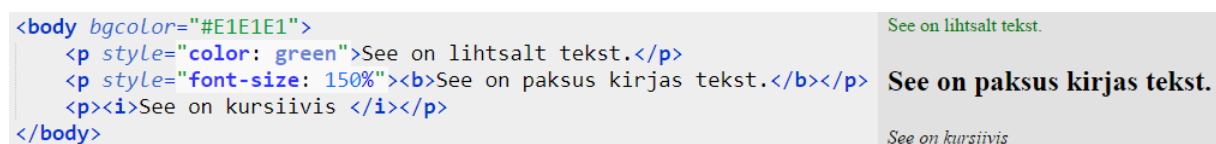
## 4 CSS-i ülevaade

CSS (*Cascading Style Sheets*) ehk kaskaadlaadistik on veebilehtede kujundamiseks mõeldud stiililehed. Laadilehed (*style sheets*) koosnevad sarnaselt hüpertexti märgistuskeelega reeglitest, mille abil brauser oskab dokumenti kasutajale kuvada. Kui HTML-i elemendid on mõeldud sisu kuvamiseks, siis CSS võeti kasutusele selleks, et neid märgendeid disainida.

CSS puhul ei ole tegemist programmeerimiskeelega, vaid koodiga, mille abil saab kirjeldada, kuidas veebilehitseja kuvab HTML-i sisu. Laadilehel saab elementide väärtustega kujundada fonte, värve, teksti, ühikuid jms [20].

Sõna „kaskaad“ CSS mõistes tähendab, et olenemata stiili asukohast, mõjutab see veebidokumenti [22]. Kui soovime kujundada veebilehte, on vaja kaskaadlaadistik siduda HTML-i dokumendiga. Selleks on kolm erinevat võimalust.

Kõige kõrgema prioriteediga on **html sisene** (*inline style*) kujundus. Sellisel juhul lisatakse stiili kood html-i elemendi sisse, kasutades **style** atribuuti (vaata joonis 19).



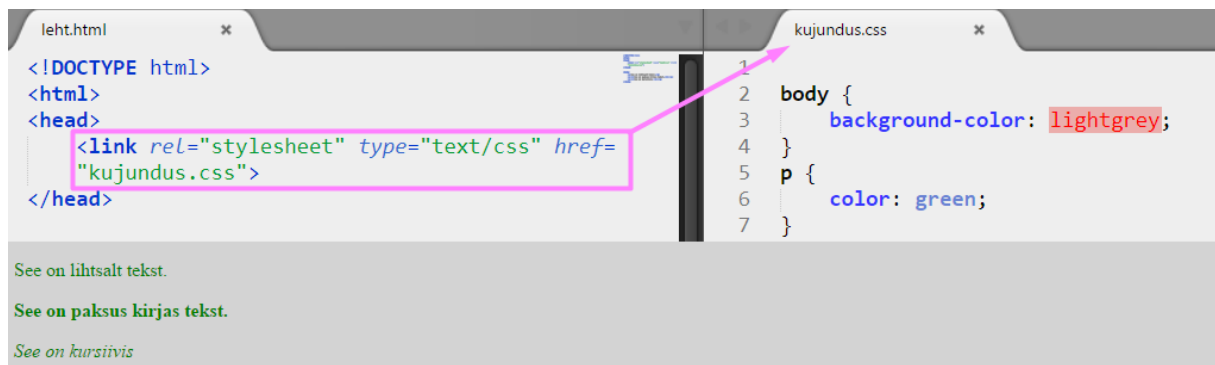
Joonis 19. html märgendite sisene kujundus.

Järgmise prioriteediga kujundus lisatakse **html lehe päisesse** (*internal style sheet*). Selleks kasutatakse **<style>** elementi **<head>** vahel (vaata joonis 20).



Joonis 20. HTML-i dokumendi sisene kujundus.

Kolmas võimalus stiili ühendada HTML-iga on **välise lehe linkimine** (*external style sheet*) **<head>** sektsioonis elemendiga **<link>** (vaata joonis 21). Selleks luuakse eraldi fail, mille laiend on **.css** (näiteks kujundus.css). Kõik, mida tahetakse kujundada, kirjutatakse loodud faili. Seda kaskaadlaadistikku saab kasutada korraga mitme erineva HTML-i dokumendi päises.



Joonis 21. Välise lehe linkimine HTML faili.

Elemendile **<link>** lisatakse mitmeid atribuute:

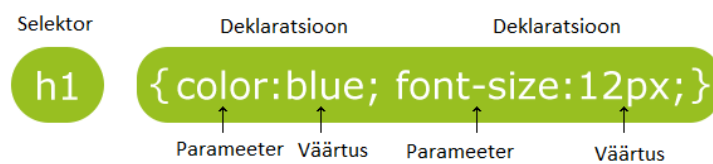
- *rel* = „stylesheet“ – määrab ära suhte praeguse ja lingitava dokumendi vahel
- *type* = „text/css“ – näitab lingitava faili MIME tüüpi (antud juhul on selleks tekst)
- *href* = „kujundus.css“ – lingitav fail

#### 4.1 CSS-i ajalugu

Aastal 1994 pakkus CSS-i idee välja H.W.Lie, mis algselt oli CHSS (*Cascading HTML Style Sheets*). 1996 andis W3C ametliku soovitusena CSS 1 versioonile. See võimaldas disainida juba tekste, värve, tabeleid jms. Seoses veebi kiire arenguga tunnustati aastal 1998 CSS 2, mis sisuliselt oli eelmise versiooni edasiarendus, kuhu lisandusid uued võimalused, nagu näiteks fikseeritud (*fixed*), absoluutne (*absoluute*) ning suhteline (*relative*) positsioneerimine, meedia mõiste, fondi laadid jms. Järgmine arendus oli CSS 2.1, kus hakati vigu parandama ning uusi võimalusi looma. Selle versiooni arendus kestis pikka aega, enne kui lõpuks aastal 2011 sai see ametliku heakskiidu W3C poolt. Järgmine arendusetapp kestab veel siiani. CSS 3 versioon jagati mooduliteks, mille kallal tehakse tööd edasi CSS 2 põhjal [21].

#### 4.2 CSS-i süntaks

CSS-i süntaks koosneb kahest suuremast osast: selektor ning deklaratsioon (vaata joonis 22).



Joonis 22. CSS-i süntaks [23].

Selektor näitab, mida soovitakse kujundada. Selleks võib olla HTML-i märgendi element, id, klass, atribuut jms. Deklaratsioon omakorda koosneb CSS-i parameetrist ning väärtusest, mis on eraldatud kooloniga. Kõik deklaratsioonid peavad lõppema semikooloniga.[23]

#### 4.2.1 Elemendi ID ja Class atribuudid

Veebidokument koosneb üldjuhul väga paljudest elementidest. Tihtipeale on vaja mingit kindlat osa või ühte elementi erinevalt kujundada. Selleks on võimalus kasutada atribuute „**class**“ ning „**id**“. Näiteks kui soovime kujundada ühel lehel erinevalt pealkirjasid, saame CSS abil kirja panna stiilid ning HTML-i elemendi sees atribuudiga **class** viitame vastavalt (vaata joonis 23). Atribuuti **id** on hea kasutada, kui soovime mingit kindlat elementi teistmoodi disainida (vaata joonis 24).



Joonis 23. Atribuut Class näide.



Joonis 24. Atribuut id näide.

Atribuudi **class** nime ei tohiks alustada numbriga, sest veebilehitseja Firefox ei toeta sellist meetodit [22].

#### 4.2.2 CSS-i koodi kommenteerimine

Kommentaare kasutatakse koodi seletamiseks ning need võivad aidata hiljem koodi lihtsamini muuta. CSS-i kommentaar algab sümbolitega **/\*** ja lõppeb sümbolitega **\*/** ning on ignoreeritud veebilehitseja poolt (vaata joonis 25) [23].



Joonis 25. CSS-i kommenteerimine.

#### 4.2.3 CSS-i parameetri väärtused

CSS-i parameetreid saab esitada kolmel erineval viisil:

- tekstiline – parameetriks on tekstiline väärtus (näiteks **font-size: medium**);

- arvuline – parameetrik on number või number ja tekst (näiteks **font-weight: 500**, **font-size: 12px**);
- värvid – värvi väärtused antakse värvikoodiga (näiteks **color: #E1E1E1**).

### 4.3 CSS-i tekstitöötlus

CSS võimaldab tekstidega ringi käia, nagu ise seda soovime. Veebilehe kujundamisel saab teha kõike: muuta kirja suurust, värvi, stiili jms. Selles peatükis annab töö autor lühikese ülevaate mõnedest võimalustest, mida CSS pakub.

#### 4.3.1 CSS-i töö tekstiga

Parameeter **text-align** võimaldab joondada teksti. Nagu MS Wordis, saab ka veebis olla joondus vasakule (*left*), paremale (*right*), keskele (*centered*) või rööpjoondus (*justified*). Selleks saab selektori deklaratsioonis anda parameetrile kokku neli väärtust. Kasutades parameetrit **text-transform**, on võimalik teksti teha suurtähtkirjaks, väiketähtkirjaks või iga sõna esimese tähe suureks. Samuti saab muuta teksti kirjastiile, teksti paksust jms. Tabel (vaata tabel 7) annab lühiülevaate mõningatest parameetritest ning nende kasutamisest.

**Tabel 7.** Mõningaid CSS-i parameetreid tekstidega töötamisel.

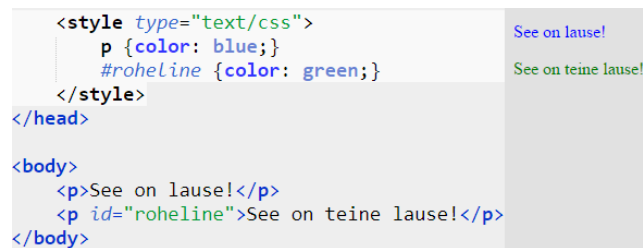
| Parameeter             | Näide ning kirjeldus  | Väärtused  |
|------------------------|---|--|
| <b>text-transform</b>  | <code>p {text-transform: uppercase;}</code><br>teksti tegemine suurtähtkirjaks                    | uppercase<br>lowercase<br>capitalize                         |
| <b>text-align</b>      | <code>p {text-align: justified;}</code><br>Teksti joondamine (rööpjoondus)                        | left<br>right<br>center<br>justified                         |
| <b>text-decoration</b> | <code>a {text-decoration: none;}</code><br>eemaldab linkidelt alakriipsu                          | none<br>underline<br>overline<br>link-throught               |
| <b>text-indent</b>     | <code>p {text-indent: 100px;}</code><br>lõigu esimesele reale taande lisamine                     | length<br>% (protsentuaalselt)                               |
| <b>line-height</b>     | <code>p {line-height: 1.2;}</code><br>määrab reavahe tekstis                                      | normal<br>numbriline väärtus<br>length<br>% (protsentuaalne) |
| <b>font</b>            | <code>p {font: 12px Calibri;}</code><br>võimaldab ühes deklaratsioonis kirjeldada kirja kujunduse |  |
| <b>font-family</b>     | <code>p {font-family: Calibri, „Times New Roman“, Times;}</code><br>määrab tekstile fondi stiili  | fondi nimi   |
| <b>font-size</b>       | <code>p {font-size: 12px;}</code><br>määrab kirja suuruse   | absoluutne suurus<br>relatiivne suurus                       |

Meeles tuleks pidada, et parameetri **font-family** kasutamisel tuleb jutumärkide vahele kirjutada kirjastiilid, mille nimes on rohkem kui üks sõna. Samuti peab soovitud stiilid kirjutama prioriteetses järjekorras. Kui veebilehitseja ei suuda kuvada esimest kirjalaadi, valib ta järgmise jne.

#### 4.3.1.1 CSS-i teksti värv

Parameeter **color** võimaldab määrata teksti värvi (vaata joonis 26). Peamiselt kasutatakse CSS-is selleks kolme erinevat viisi väärtuse andmiseks:

- värvi nimi – näiteks „**blue**“;
- HEX väärtust – näiteks „**#0000FF**“;
- RGB väärtus – näiteks „**rgb(0, 0, 255)**“.



Joonis 26. Värvide määramine tekstile.

HEX ning RGB väärtusi toetavad kõik populaarsemad veebilehitsejad ning nendega saab määrata kõiki värvitoone.

#### 4.3.2 CSS-i tabelid

CSS võimaldab anda tabelile välimuse. Saame muuta äärisi, värve, suurust, lisada efekte jms. HTML-is lisatud tabel on ilmetu ning kujundamata ja tihtipeale raskesti loetav. Olgu meil lihtne tabel, mida soovime kujundada (vaata joonis 27). CSS võimaldab ka eraldi elementide kaupa tabelit disainida, aga antud peatükis tegeleme tabeliga üldiselt.

```
/*Tabeli lahtritele ja päistele äärise loomine*/
td, th { border: 1px black solid; }
/*Lahtrite ühendamine omavahel, et iga lahter ei oleks eraldi kujundatud*/
table { border-collapse: collapse; }
/*Lisame tabeli päisele taustavärvi*/
thead {background: orange; }
/*Lisame tabeli sisule tasutavärvi*/
tbody {background: aqua;}
```

Määrame ära, milliseid elemente me soovime kujundada. Antud koodi näites määrame kõikide tabeli osadele mustad ning ühtlase joonega äärised paksusega 1 piksel. Värvime eraldi tabeli päise ning keha osa.

| Kujundamata HTML tabel |      | Kujundatud HTML tabel |      |
|------------------------|------|-----------------------|------|
| Nimi                   | Auto | Nimi                  | Auto |
| Mart Juurikas          | Opel | Mart Juurikas         | Opel |
| Jaana Jānes            | Ford | Jaana Jānes           | Ford |

Joonis 27. HTML tabeli kujundamine.

CSS abil saab tabeli taustaks määrata ka pilti, aga sel juhul peaks jälgima, et tekst ei jääks märkamatuks. Kasutades **:hover** näiteks märgendi **<tr>** peal, saame lisada tabelile efekti, et liikudes hiirega üle tabeli, muutub rea värv: **tr:hover {background-color: #f5f5f5}**.

#### 4.4 CSS-i kursorid

Veebilehel on võimalik muuta ka kasutaja kursori väljanägemist (vaata joonis 28). Selleks on CSS-is kasutusel süntaks parameeter **cursor**. Kursori väljanägemist saab muuta erinevate elementide peal eraldi.

```
/*Kursori stiili muutmine*/
cursor: value;
```

|              |                 |               |              |
|--------------|-----------------|---------------|--------------|
| I auto       | ↕ move          | 🖱 no-drop     | ↕ col-resize |
| 🖱 all-scroll | 🖱 pointer       | 🚫 not-allowed | ↕ row-resize |
| + crosshair  | 🖱 progress      | ↔ e-resize    | ↗ ne-resize  |
| 🖱 default    | I text          | ↑ n-resize    | ↖ nw-resize  |
| 🖱 help       | ↕ vertical-text | ↓ s-resize    | ↘ se-resize  |
| I inherit    | ⌛ wait          | ↔ w-resize    | ↙ sw-resize  |

Joonis 28. CSS kursorid[24].

Erinevaid kursoristiile on palju. Mõnikord võib juhtuda, et veebilehitseja seda ei toeta ning ei oska kuvada. Sellisel juhul lihtsalt kaob see efekt ära ning stiil on asendatud tavapärasega.

#### 4.5 CSS-i pseudoklassid ning -elemendid

Pseudoklasse kasutatakse mingite efektide (vaata joonis 29) loomiseks selektoritele või nende osadele. Näiteks liikudes hiirega üle mingi kindla elemendi sisu, linkide eristamiseks või mingi osa fokuseerimiseks.

```
/*Pseudoklassi kasutamine*/
selektor: pseudoklass { tegevus }
```

Tabelis (vaata tabel 8) on välja toodud mõned levinumad pseudoklassid ning -elemendid.

**Tabel 8.** Levinud pseudoklassid ja –elemendid.

| Selektor              | Näide           | Kirjeldus   |
|-----------------------|-----------------|---|
| <b>:hover</b>         | a:hover         | Kursoriga üle elemendi liikumine                      |
| <b>:focus</b>         | input:focus     | <b>&lt;input&gt;</b> elemendi fokuseerimine           |
| <b>:link</b>          | a:link          | Küllastamata link                                     |
| <b>:visited</b>       | a:visited       | Küllastatud link                                      |
| <b>::after</b>        | p::after        | Lisab sisu pärast <b>&lt;p&gt;</b> elementi           |
| <b>::before</b>       | p::before       | Lisab sisu enne <b>&lt;p&gt;</b> elementi             |
| <b>::first-letter</b> | p::first-letter | Vormindab <b>&lt;p&gt;</b> elemendi sisu esimese tähe |

```

/* küllastamata link */
a:link {
    color: red;
}
/* küllastatud link */
a:visited {
    color: green;
}
/* kursoriga üle liikudes link */
a:hover {
    color: yellow;
}
/* valitud link */
a:active {
    color: blue;
}

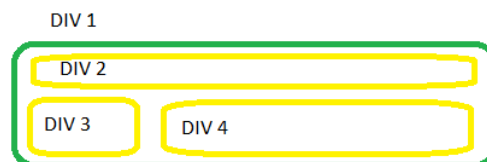
```

**Joonis 29.** Linkide värvide muutus vastava tegevuse korral.

Tähelepanu tasub pöörata sellele, et pseudoklasside kasutamisel on oluline järjekord.

#### 4.6 CSS-iga positsioneerimine

CSS võimaldab elemente positsioneerida. Algselt paikneb sisu veebilehel sellises järjekorras, nagu HTML-i dokumendis on kirja pandud. Sisuliselt näeb veebileht välja nagu üks suur konteiner, kuhu sisse on paigutatud palju „kaste“ (vaata joonis 30). Tihtipeale soovitakse aga neid ümber paigutada ning teistmoodi kuvada. Selleks on loodud atribuut **position**, mis pakub positsioneerimiseks nelja erinevat varianti: **static**, **relative**, **fixed** ning **absolute**. Lisaks kasutatakse asukoha määramise parameetreid (välja arvatud **static** puhul) „**top**“ (kaugus ülevalt), „**bottom**“ (kaugus alt), „**left**“ (kaugus vasakult) ja „**right**“ (kaugus paremalt), mis määravad objekti asukoha.



**Joonis 30.** Veebileht kui kastimudel.

Staatiline ehk **static** on vaikimisi väärtus ning elemendid paigutatakse üksteise alla.



```
/*Static kasutamine*/  
div.static { position: static; }
```

Suhteline positsioneerimine ehk **relative** võimaldab objekti liigutada tema esialgse asukoha suhtes. Kasutades parameetrit „**top**“, määratakse ära, kui palju liigutatakse objekti tema esialgsest kohast allapoole, „**bottom**“ ülespoole, „**left**“ kui palju paremale ning „**right**“ paremale.

```
/*Static kasutamine*/  
div.static { position: relative; }
```

Absoluutne positsioneerimine ehk **absolute** ei sõltu teistest elementidest. Asukoha määramise parameetrite abil määratakse ära objekti paiknemine lehel. CSS-i abil öeldakse, kui kaugel see asub ülevalt, alt, paremast või vasakust äärest.

```
/*Absolute kasutamine*/  
div.static { position: absolute; }
```

Fikseeritud positsioneerimine ehk **fixed** on kindla suuruse ja asukohaga objekt. See objekt määratakse ära lehekülje põhjal ehk brauseri akna suhtes, mitte kasti järgi, mille sees see asub. Fikseeritud positsiooni puhul ei liigu objekt kaasa lehe kerimisel.

```
/*Absolute kasutamine*/  
div.static { position: fixed; bottom: 0; width: 200px;}
```

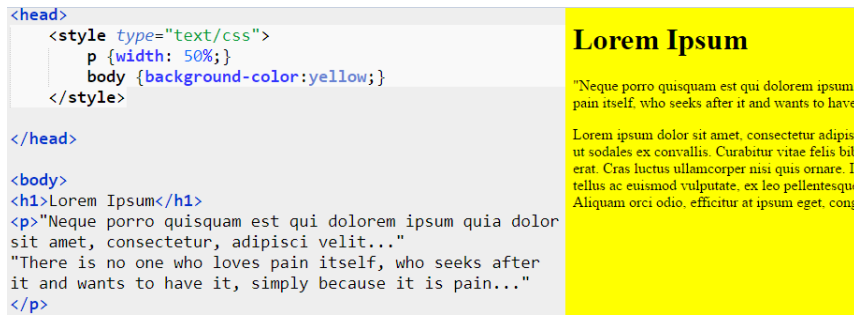
Kasutades **z-index** parameetrit, on võimalus määrata elementide kuvamise järjekorda. Suurema väärtusega kuvatakse kõige ees. Oluline on meeles pidada, et kui soovitakse järjekorda muuta, peab positsioneerimine olema tehtud absoluutse parameetriga [25].

#### 4.7 CSS-i taustad

Veebilehele annab suure osa ilmast taust. See peaks olema piisavalt kujundatud, et kasutajale kuidagi meelde jääda. CSS võimaldab veebilehele tausta määrata mitmel moel.

Kõige lihtsam moodus tausta määramiseks on lehele anda ühtlane täide (vaata joonis 31).

```
/*Tausta värv*/  
body {background-color: yellow;}
```



Joonis 31. Ühtlane täide.

Teine variant on kasutada üleminekuvärve (*gradient*). Astmikvärvid võimaldavad tausta kujundada nii, et ühelt värvilt minnakse sujuvalt üle teistesse värvitoonidesse. Varem tuli sellise efekti loomiseks kasutada pilte, mistõttu võttis veebilehe laadimine kauem aega ning kulutas liigselt ressursse. Kuna üleminekuvärvid genereerib veebilehitseja, siis on ka veebilehe laadimise aeg kiirem ning suurendamisel ei teki anomaaliaid. Kuna kõik brauserid veel ei toeta sellise efekti loomist, siis tuleb alati defineerida ka tavaline taust nagu näiteks ühtlane täide. CSS-is on kasutusele võetud kahte tüüpi astmikvärve:

- Lineaarne (*linear gradient*)
- Radiaalne (*radial gradient*)

Lineaarne üleminekuvärv liigub alla, üles, paremale, vasakule või diagonaalselt. Selleks tuleb defineerida vähemalt kaks värvi. Erinevad veebilehitsejad kasutavad erinevaid standardeid, seega tuleb ka erinevaid parameetreid kasutada (vaata joonis 32).

```

/*Tausta linear-gradient*/
body {background: linear-gradient(suund, värv1, värv2, ...);}

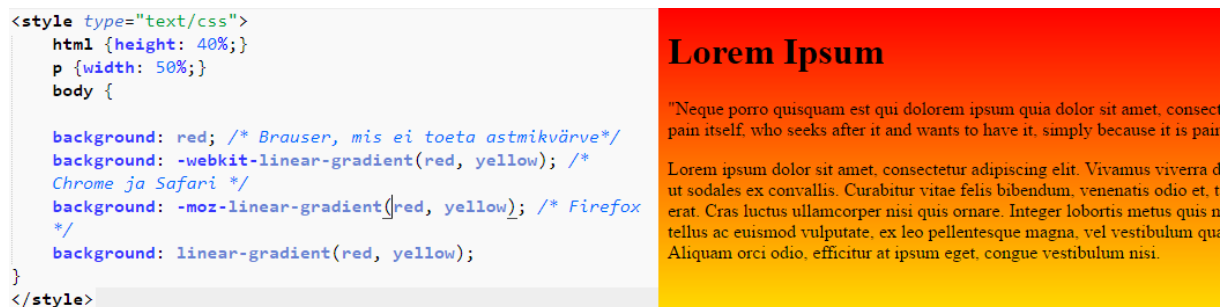
```

Radiaalne üleminekuvärv on sarnane lineaarsele, aga võimaldab lisada rohkem parameetreid. Vaikimisi on kujundiks ring ning hakatakse värvima keskelt (vaata joonis 33).

```

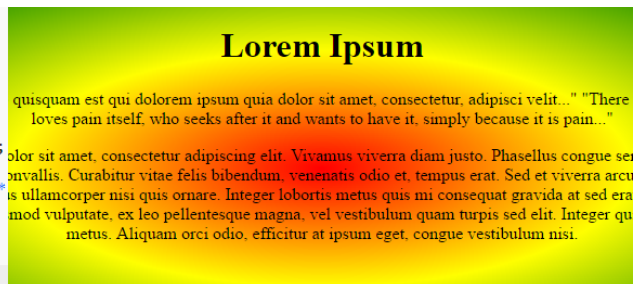
/*Tausta radial-gradient*/
body {background: radial-gradient(kujundi määramine, värv1, värv2, ...);}

```



Joonis 32. Lineaarne astmikvärvi ülevaade.

```
<style type="text/css">
  html {height: 40%;}
  p {width: 50%;}
  body {
    background: red; /* Brauser, mis ei toeta astmikvärve*/
    background: -webkit-radial-gradient(red, yellow, green);
    /* Chrome ja Safari */
    background: -moz-radial-gradient(red, yellow, green); /*
    Firefox */
    background: radial-gradient(red, yellow, green);
  }
</style>
```



Joonis 33. Radiaalne astmiktäide.

Veel üks viis tausta määrata on kasutada taustapilti. CSS võimaldab meil panna taustapilt kordusesse, et kulutada vähem ressursse selle kuvamisel. Selleks luuakse tihtipeale pildifailina muster (näiteks **.gif** või **.jpg** fail), mida soovitakse kuvada (vaata joonis 34).

```
body {
  background-image: url(muster.gif); /*taustapildi linkimine*/
  background-repeat: repeat; /*taustapildi kordamine*/
}
```



Joonis 34. Mustri kordamine ja mitte kordamine veebilehel.

Atribuudile **background-repeat** saab anda neli väärtust:

- **repeat** – vaikimisi brauseri poolt, korratakse üle lehe;
- **no-repeat** – taustapilti ei korrata;
- **repeat-x** – taustapilti korratakse ülevalt alla;
- **repeat-y** – taustapilti korratakse vasakult paremale.

## Kokkuvõte

Käesoleva lõputöö eesmärgiks oli luua ülevaatlik veebilehe loomise kursus algajatele. Materjalide koostamisel on arvesse võetud, et sihtgrupiks on gümnaasiumi õpilased, kellel on huviõppida veebidokumentide koostamist. Töö jaguneb kolmeks osaks.

Esimeses osas antakse ülevaade loodava kursuse ülesehitusest ning eesmärkidest. Kirjeldatakse, milliseid materjale autor kasutab ning kuidas neid õpilastele esitada. Esimeses peatükis on lühidalt kirjeldatud, kuidas õpilasi hinnata, et neil oleks rohkem motivatsiooni kursust läbida.

Teine osa annab väikese ülevaate HTML-i keelest ning lühidalt autori arvates mõnedest olulistest võimalustest. Keskendutud on pigem näidetele, et oleks aru saada, mida vastav skriptirida reaalselt teeb.

Kolmandas osas on samuti ülevaade antud mõnedest CSS võimalustest, et disainida lihtne veebileht. Sarnaselt teisele osale on kasutatud pilte, et paremini mõista, mida erinevad koodiread muudavad veebilehe kujundust.

Eesmärgiks oli ka õpilastes huvi äratamine. Veebilehtede loomise kogemus on tänapäeval infotehnoloogia sektoris kasulik. Töötades koolis õpetajana, julgen oma kogemuste põhjal väita, et huvi matemaatika ja teiste reaalainete vastu on noorte seas pigem kahanemas ning soov antud valdkonda minna õppima kõrgkooli pigem väiksem.

Koostöös erinevate koolide õpetajate ja õpilastega olen teada saanud, et paljudes koolides õpetatakse peamiselt rakendusprogramme (*MS Word*, *MS Excel*, *MS Access*) ning vähesel määral ka programmeerimist. Veebilehtede loomise algkursus annab õpilastele kindlasti parema ülevaate infotehnoloogia valdkonnast.

## Viited

- [1] Teder, Merike 2012. „Aaviksoo loetles eelseisva haridusreformi põhipunktid“. *Postimees*, 11. aprill, URL <http://www.postimees.ee/804482/aaviksoo-loetles-eelseisva-haridusreformi-pohipunktid-viimati-vaadatud-15.03.2016>
- [2] W3Schools. [Online]. <http://www.w3schools.com/> - viimati vaadatud 08.04.2016
- [3] Codecademy. [Online]. <https://www.codecademy.com/> - viimati vaadatud 08.04.2016
- [4] Moodle.net. Moodle Statistics. [Online]. <https://moodle.net/stats/?lang=et> – viimati vaadatud 08.04.2016
- [5] Pilt, Lehti 2016. E-õppe statistika 2015. – Tartu Ülikooli e-õppe ajakiri, 4. veebruar. <http://etu.ut.ee/kevad-2016/statistika-2015/> - viimati vaadatud 08.04.2016
- [6] Tartes, Tõnis 2010. Mis on Moodle?. – Tartu Ülikooli e-õppe ajakiri, 21. märts. <http://etu.ut.ee/kevad-2010/mis-on-moodle/> - viimati vaadatud 08.04.2016
- [7] Tartu Ülikooli „Õppijate hindamine Moodle’is“ juhend – <https://wiki.ut.ee/pages/viewpage.action?pageId=17114079> - viimati vaadatud 11.04.2016
- [8] Rinde Andrus 2015. Veebilehtede loomine, html5 [http://www.cs.tlu.ee/~rinde/oppetoo/2015/www/pdf/prax\\_1\\_html5\\_oigekiri\\_link\\_pilt.pdf](http://www.cs.tlu.ee/~rinde/oppetoo/2015/www/pdf/prax_1_html5_oigekiri_link_pilt.pdf) - viimati vaadatud 12.05.2016
- [9] PyramDesign. Html Sissejuhatus. [Online]. [http://pyramdesign.angelfire.com/html\\_sissejuhatus.html](http://pyramdesign.angelfire.com/html_sissejuhatus.html) - viimati vaadatud 12.05.2016
- [10] W3Schools. HTML Introduction. [Online]. [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp) - viimati vaadatud 20.04.2016
- [11] Heikki Vallaste. E-teatmik. [Online]. <http://www.vallaste.ee/index.asp> - viimati vaadatud 12.05.2016
- [12] W3Schools. HTML <head> tag. [Online]. [http://www.w3schools.com/tags/tag\\_head.asp](http://www.w3schools.com/tags/tag_head.asp) - viimati vaadatud 21.04.2016
- [13] W3Schools. HTML text Formatting Elements. [Online]. [http://www.w3schools.com/html/html\\_formatting.asp](http://www.w3schools.com/html/html_formatting.asp) - viimati vaadatud 22.04.2016
- [14] W3Schools. HTML Headings. [Online]. [http://www.w3schools.com/html/html\\_headings.asp](http://www.w3schools.com/html/html_headings.asp) - viimati vaadatud 22.04.2016
- [15] W3Schools. HTML Lists. [Online]. [http://www.w3schools.com/html/html\\_lists.asp](http://www.w3schools.com/html/html_lists.asp) - viimati vaadatud 23.04.2016
- [16] W3Schools. HTML Forms. [Online]. [http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp) - viimati vaadatud 24.04.2016
- [17] Metshein, Mario. HTML5 – Vorm ja selle elemendid I. [Online]. <http://metshein.com/index.php/veeb/html5/518-15-html5-vorm-ja-selle-elemendid-i> - viimati vaadatud 03.04.2016

- [18] W3Schools. HTML Tables. [Online]. [http://www.w3schools.com/html/html\\_tables.asp](http://www.w3schools.com/html/html_tables.asp) - viimati vaadatud 04.05.2016
- [19] Metshein, Mario. HTML5 – Tabelid. [Online]. <http://metshein.com/index.php/veeb/html5/517-14-html5-tabelid> - viimati vaadatud 06.05.2016
- [20] Heikki Vallaste. E-teatmik. [Online]. <http://www.vallaste.ee/index.asp> - viimati vaadatud 26.04.2016
- [21] Wikipedia. (2016, Mai). Cascading Style Sheets. [Online]. [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets#History](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#History) - viimati vaadatud 27.04.2016
- [22] PyramDesign. Html Sissejuhatus. [Online]. [http://pyramdesign.angelfire.com/css\\_sissejuhatus.html](http://pyramdesign.angelfire.com/css_sissejuhatus.html) - viimati vaadatud 27.04.2016
- [23] W3Schools. CSS Syntax and Selectors. [Online]. [http://www.w3schools.com/css/css\\_syntax.asp](http://www.w3schools.com/css/css_syntax.asp) - viimati vaadatud 27.04.2016
- [24] Kourbatov, Alexei. Mouse Cursor Styles. [Online]. <http://www.javascripter.net/faq/styleesc.htm> - viimati vaadatud 07.05.2016
- [25] MetsheinMario. CSS3 – Sisu paigutus (positsioneerimine). [Online]. <http://metshein.com/index.php/veeb/css3/537-14-css3-sisu-paigutus-positsioneerimine> - viimati vaadatud 08.05.2016
- [26] Vikiõpikud. Kategooria: HTML. [Online]. <https://et.wikibooks.org/wiki/Kategooria:HTML> - viimati vaadatud 09.05.2016
- [27] „Matemaatikast saab gümnaasiumi uue õppekava kardetuim aine“(2012). *Postimees*, 11. september, URL <http://www.postimees.ee/968240/matemaatikast-saab-gumnaasiumi-uae-oppekava-kardetuim-aine>

## Litsents

### **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, **Martin Merisalu** (sünnikuupäev: 07.12.1990),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose  
**Veebilehe loomise e-kursus gümnasistidele**,

mille juhendaja on **Ljubov Jaanuska**,

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **12.05.2016**